In the Specification

Please replace the paragraph beginning at line 4 on page 15 with the following marked up paragraph:

A1

-- In the embodiments shown in Figures 3A-D, the dictionary file also contains an optional count of the number of occurrences of each unique value in the column. By storing a count of the number of occurrences of each value in the dictionary file, the popular SUM, COUNT, AVERAGE, MIN and MAX aggregate functions can be computed much more quickly, since the data file does not need to be scanned, and at minimal space overhead, especially if a space-efficient representation for the counters is used, e.g., a variable-length integer. Furthermore, the count of occurrences allows the system to ~~exit~~ execute an equivalence query without having to scan all entries within a data file. For example, assume the query is "select *x* where *y* = *z*" and the dictionary file for the data file for column *y* indicates that only two occurrences exist. Therefore, once the two occurrences have been found, the scan can be terminated.--

Please replace the paragraph beginning at line 10 on page 19 with the following marked up paragraph:

A2

-- In one embodiment, when chronological data is stored in the system by appending multiple rows in a single batch, an extra column is used to record a batch "upload" identifier and provide the ability to recover from an erroneous batch store without affecting other data uploaded in different batches. For example, suppose there are less than 65536 uploads in a time range, then the variable-length integer encoding described above will require no more than 6 bits per a 250 byte record to encode the batch identifier. This will compress to approximately less than 1 bit per record, a trivial amount compared to the 10-15 bytes per record post-compression. It will be appreciated that number of uploads and record size vary with the type of data.--

Please replace the paragraph beginning at line 25 on page 22 with the following marked up paragraph:

A3

-- When data is replicated across storage devices as shown in Figure 4A, an aggregation service will receive duplicate results from the parallel processing of a sub-query on all the filtering services. For example, aggregation service ~~103~~ 403 will receive the same results for data set B 415 on filtering service 409 as from data set B 417 on filtering service 411.

A3

The aggregation service ~~103~~ 403 should detect the duplicates and use only one of the data sets. In one embodiment, each duplicate data set is stored under a different name, such as nodeX.datasetname and nodeY.datasetname, and the aggregation service uses the name to determine duplicate results.--

Please replace the paragraph beginning at line 16 on page 29 with the following marked up paragraph:

A4

-- Turning now to Figure 6, the acts to be performed by computers when processing a query are shown. Upon receipt of a SQL query or sub-query at a query master, compute or storage node, the query processing method 600 illustrated in Figure 6 is invoked. The method 600 parses the query into a tree of expressions representing the semantics of the query (block 601) and performs preliminary checks to ensure the query is semantically valid (block 603). All references to columns are resolved (block 605) and ~~the method 600~~ a series of corresponding column I/O streams are initialized, one per column (block 607). The processing represented by block 607 initializes different types of I/O streams based on the type of node executing the method 600. For instance, the column I/O streams used by a storage node obtain the data from the column-files stored on that node, while the column I/O streams used by the query master and compute nodes obtain the data from other compute and storage nodes through the network as previously described. When the method 600 is executed by the lowest level compute nodes and the requested data is fully or partially mirrored, the processing represented by block 607 determines whether to initialize I/O streams to obtain data from all the mirroring storage nodes in parallel or from only a subset of the mirroring storage nodes.--

Please replace the paragraph beginning at line 9 on page 34 with the following marked up paragraph:

A5

-- Figure 8B shows one example of a conventional computer system that can be used as a client computer system or a server computer system or as a web server system. It will also be appreciated that such a computer system can be used to perform many of the functions of an Internet service provider, such as ISP 5. The computer system 51 interfaces to external systems through the modem or network interface 53. It will be appreciated that the modem or network interface 53 can be considered to be part of the computer system 51. This interface 53 can be an analog modem, ISDN modem, cable

A5

modem, token ring interface, satellite transmission interface (e.g. "Direct PC"), or other interfaces for coupling a computer system to other computer systems. The computer system 51 includes a processing unit 55, which can be a conventional microprocessor such as an Intel Pentium microprocessor or Motorola Power PC microprocessor. Memory 59 is coupled to the processor 55 by a bus 57. Memory 59 can be dynamic random access memory (DRAM) and can also include static RAM (SRAM). The bus 57 couples the processor 55 to the memory 59 and also to non-volatile storage 65 and to display controller 61 and to the input/output (I/O) controller 67. The display controller 61 controls in the conventional manner a display on a display device 63 which can be a cathode ray tube (CRT) or liquid crystal display. The input/output devices 69 can include a keyboard, disk drives, printers, a scanner, and other input and output devices, including a mouse or other pointing device. The display controller 61 and the I/O controller 67 can be implemented with conventional well known technology. A digital image input device ~~61~~ 71 can be a digital camera which is coupled to an I/O controller 67 in order to allow images from the digital camera to be input into the computer system 51. The non-volatile storage 65 is often a magnetic hard disk, an optical disk, or another form of storage for large amounts of data. Some of this data is often written, by a direct memory access process, into memory 59 during execution of software in the computer system 51. One of skill in the art will immediately recognize that the term “computer-readable medium” includes any type of storage device that is accessible by the processor 55 and also encompasses a carrier wave that encodes a data signal.--